A new algorithm for Fixed Design Regression and Denoising

F. Comte *(Corresponding author)*

MAP5, FRE CNRS 2428,
Université Paris 5,
45 rue des Saints-Pères,
75270 PARIS cedex 06,
FRANCE.
Tel. 01-42-86-21-18, Fax. 01-42-49-37-12.
e-mail: comte@biomedicale.univ-paris5.fr

Y. Rozenholc

INRA and LPMA, UMR CNRS 7599, Université Paris VII and Université du Maine, Le Mans, FRANCE. e-mail: rozen@math.jussieu.fr

Summary. In this paper, we present a new algorithm to estimate a regression function in a fixed design regression model, by piecewise (standard and trigonometric) polynomials computed with an automatic choice of the knots of the subdivision and of the degrees of the polynomials on each sub-interval. First we give the theoretical background underlying the method: the theoretical performances of our penalized least square estimator have been studied at length in other papers and are based on non-asymptotic evaluations of a mean-square type risk. Then we explain how the algorithm is built and possibly accelerated (to face the case when the number of observations is great), how the penalty term is chosen and why it contains some constants requiring an empirical calibration. Lastly, a comparison with some well-known or recent wavelets methods is led: this brings out that our algorithm behaves in a very competitive way in term of denoising and of compression.

Keywords. Least square regression. Piecewise polynomials. Adaptive estimation. Model selection. Dynamical programmation. Algorithm for denoising.

1. Introduction

We consider in this paper the problem of estimating an unknown function f from $[0,1]$ into $\mathbb{R}$ when we observe the sequence Y_i, $i = 1, \ldots, n$, satisfying

$$Y_i = f(x_i) + \sigma\varepsilon_i, \tag{1}$$

for fixed $x_i, i = 1, \ldots, n$ in $[0,1]$ with $0 \leq x_1 < x_2 < \ldots < x_n \leq 1$. Most of the theoretical part of the work concerns any type of design but only the equispaced design $x_i = i/n$ is computationally considered and implemented. Here $\varepsilon_i, 1 \leq i \leq n$ is a sequence of independent and identically distributed random variables with mean 0 and variance 1. The positive constant σ is first assumed to be known. Extensions to the case where it is unknown are proposed.

We aim at estimating the function f using a data driven procedure. In fact, we want to approximate f by piecewise standard and trigonometric polynomials in a spirit analogous but more general than e.g. Denison et al. (1998). We also want to choose among "all possible subsets of a large collection of pre-specified candidates knot sites" as well as among various degrees on each subinterval defined by two consecutive knots.

Our method is based on recent theoretical results obtained by Baraud (2000); Baraud (1998), Baraud et al. (2001a,b) who adapted to the regression problem general methods of model selection and adaptive estimation initiated by Barron and Cover (1991) and developed by Birgé and Massart (1998), Barron et al. (1999). Results on Gaussian regression can also be found in Birgé and Massart (2001). Of course, the literature on the subject of fixed design regression did not start at that time but all the results we have in mind have the specificity of giving non asymptotic risk bounds and of dealing with adaptive estimators. The first results about adaptation in the minimax sense in that context were given by Efromovich and Pinsker (1984). Some asymptotic results have been also proved by Shibata (1981), Li (1987), Polyak and Tsybakov (1990). An overview of most nonparametric techniques is also given by Hastie and Tibshirani (1990). Note that from the theoretical point of view, most results can be used to do some variable selection in an additive model for instance (i.e. when f depends on several variables and $f(x) = f(x^{(1)}, \ldots, x^{(p)}) = f_1(x^{(1)}) + \ldots + f_p(x^{(p)})$), but this point is not empirically studied here: in particular, the multivariate extensions of the algorithm would probably require some more work. Lastly, it is worth mentioning that most available algorithms deal with equally spaced design; results and proposals concerning the more general case of a non necessarily equi-spaced design are quite recent. Some of them can be found for instance in Antoniadis and Pham (1998), see also the survey in Antoniadis et al. (2002).

An attractive feature of the method which is developed here is that, once a calibration step is done, everything is completely automatic and quite fast. Friedman and Silverman (1989) already gave an algorithm for optimizing over the number and location of the knots of the partition in an adaptive way: this algorithm is used by Denison et al. (1998) but the later calibrate a piecewise cubic fit. In other words, all their polynomials have the same degree a priori fixed to be 3. Ours have variable degrees between 0 and r_{max} (which is $r_{max} = 75$ in experiments) and those degrees are also automatically chosen. This is an important flexibility, for denoising square signals for instance. Moreover, the calibration operation being done once for all, the only input of the algorithm are the maximal number of knots to be considered and the maximal degree r_{max}. We do not have any complicated or arbitrary stopping criterium to deal with, as in many MCMC methods, we do not have any problems of initialization either. A great number of wavelet methods have also been recently proposed in the literature. For an exhaustive presentation and test of these methods, the reader is referred to Antoniadis et al. (2002). Therefore, we compared our method with standard toolboxes implemented by Donoho and Johnstone (1994), as well as with some of the more recent methods tested in Antoniadis et al. (2002). Note that we found out that Coifman and Donoho (1995)'s improvement of Coifman and Wirkhauser (1992)'s method was the best competitor. The performances of our algorithm prove that our method is very good , for any sample size, any type of function f, and whether σ is known or not. Let us mention also that our method seems to globally behave in a very competitive way, in term of $\mathbf{L}_2$-error performances as well as in term of compactness of the representation of

the signal. Besides, we deal with much more general frameworks. Our main drawback until now is in term of the complexity of our algorithm,which is of order $O(n^2)$ linear operations or $O(n^3)$ elementary operations $(+, \times, <)$ when theirs is of order $O(n \log_2(n))$ elementary operations. Actually we propose a quick but approximated version with complexity of order $O(n)$ linear operations or $O(n^2)$ elementary operations $(+, \times, <)$. As a counterpart their analysis includes $2^{n/2}$ bases which are constructed on dyadic partitions whereas ours includes about $(2r_{max})^n$ different basis which are constructed on general partitions.

Section 2 gives some more details on the theoretical part of the procedure. We present first its formal principle. Then a theoretical result is stated as well as possible extensions and consequences. Finally, the general form of the penalty we are working with is written. In Section 3, details about how the estimate is computed are given, two relevant bases are described (one for the space of standard polynomials, the other for the space of trigonometric polynomials) and the reason for the choice of the form of the penalty term involved in the computation of the estimate is explained. Section 4 presents the algorithm: the two main ideas, namely localization and dynamical programming are developed. The scheme for accelerating the algorithm without loosing its good properties is introduced. Section 5 presents the empirical results for both the complete algorithm and the accelerated algorithm. The calibration procedure is led with the complete algorithm. Then both methods are compared (in term of $\mathbf{L}_2$-error and of compression performances) with wavelet denoising developed by Donoho and Johnstone (1994) and Donoho et al. (1995) whose toolbox is available on Internet with test functions that we also used. Comparison results with 8 other recent methods are also provided. Lastly Section 6 gives some concluding remarks.

2. The general method

2.1. General framework

We aim at estimating the function f of model (1) using a data driven procedure. For that purpose we consider families of linear spaces generated by piecewise polynomials bases and we compute for each space (base) the associated least square estimator. Our procedure chooses among the resulting collection of estimators the "best" one, in a sense that will be precised. The procedure is the following. Let D_{max} and r_{max} be two fixed integers and D an integer such that $0 \leq D \leq D_{max}$. For any D, we choose a partition of $[0,1]$, that is a sequence $a_1, \ldots, a_{D-1}$ of $D-1$ real numbers in $[0,1]$ such that $0 = a_0 < a_1 < \ldots < a_{D-1} < a_D = 1$, and a sequence of degrees, that is integers $r_1, \ldots, r_D$, such that for any $d, 1 \leq d \leq D$, $0 \leq r_d \leq r_{max}$. Then, denoting by

$$m = (D, a_1, \ldots, a_{D-1}, r_1, \ldots, r_D) \tag{2}$$

we define a linear space S_m as the set of functions g defined on $[0,1]$ that admit the following kind of decomposition: let $I_d = [a_{d-1}, a_d[$ for $d = 1, \ldots, D-1$, and $I_D = [a_{D-1}, a_D]$, then

$$g(x) = \sum_{d=1}^{D} P_d(x) \mathbb{1}_{I_d}(x), \quad P_d \text{ polynomials with degree } r_d, d = 1, \ldots, D.$$

We define $\ell(I)$ as the number of x_k falling in the subinterval I and we call it "length of I".

The space S_m generated in this way has the dimension $D_m = \sum_{i=1}^{D}(r_i+1)$. If we call $\mathcal{M}_n \subset \{1,...,D_{max}\} \times [0,1]^D \times \{0,\dots,r_{max}\}^D$ a finite set of all possible choices for m, the family of linear spaces of interest is then $\{S_m, m \in \mathcal{M}_n\}$.

Given some m in $\mathcal{M}_n$, we define the standard least square estimator $\hat{f}_m$ of f in S_m by

$$\sum_{i=1}^{n}(Y_i - \hat{f}_m(x_i))^2 = \min_{g\in S_m}\sum_{i=1}^{n}(Y_i - g(x_i))^2. \tag{3}$$

In other words, we compute the minimizer $\hat{f}_m$ for all g in S_m of the contrast $\gamma(g)$ where

$$\gamma(g) = \frac{1}{n}\sum_{i=1}^{n}\left[Y_i - g(x_i)\right]^2. \tag{4}$$

Each model m being associated with an estimator $\hat{f}_m$, we have a collection of estimators $\{\hat{f}_m, m \in \mathcal{M}_n\}$ and we look for a data driven procedure $\hat{m} = \hat{m}(Y_i, i = 1,\dots,n)$ which selects automatically among the set of estimators the one that is defined as *the* estimator of f:

$$\tilde{f} = \hat{f}_{\hat{m}}.$$

$\hat{m}$ is a triple "number of bins, partition, degree of the polynomials on each piece" with values in $\mathcal{M}_n$, $(\hat{D}, (\hat{a}_1,\dots,\hat{a}_{\hat{D}-1}), (\hat{r}_1,\dots,\hat{r}_{\hat{D}}))$ based solely on the data and not on any a priori assumption on f.

Let us precise what the "best" estimator is and the way to select it. We measure the risk of an estimator via the expectation of some random $\mathbf{L}_2$-norm. If $\hat{f}$ is some estimator of f, the risk of $\hat{f}$ is defined by

$$\mathbf{E}\left(\frac{1}{n}\sum_{i=0}^{n}(f(x_i) - \hat{f}(x_i))^2\right) := d_n^2(f,\hat{f}),$$

since $\hat{f}$ is random through its dependency on the Y_i's. The risk of $\hat{f}_m$, where $\hat{f}_m$ is an estimator built as in relation (3), can in fact be proved to be equal (see equation (2) in Baraud (2000)) to

$$d_n^2(f,S_m) + \frac{\dim(S_m)}{n}\sigma^2$$

where $d_n(f,S_m) = \inf_{t\in S_m} d_n(f,t)$ and $\dim(S_m)$ denotes the dimension of S_m. Therefore an ideal selection procedure choosing $\hat{m}$ should look for an optimal trade-off between $d_n^2(f,S_m)$, the so-called bias term and $\sigma^2\dim(S_m)/n$, the so-called variance term . In other words, we look for a model selection procedure $\hat{m}$ such that the risk of the resulting estimator $\hat{f}_{\hat{m}}$ is almost as good as the risk of the best least squares estimator in the family. More precisely, our aim is to find $\hat{m}$ such that

$$d_n^2(f,\hat{f}_{\hat{m}}) \le C \min_{m\in\mathcal{M}_n}\left\{d_n^2(f,S_m) + \sigma^2\frac{L_m\dim(S_m)}{n}\right\}, \tag{5}$$

where the L_m's are some weights related to the collection of models $\{S_m, m \in \mathcal{M}_n\}$. This inequality means that, up to a constant C (which has to be not too far from one for the

result to be of some interest) our procedure chooses an optimal model and inside that model an optimal estimator in the sense that it realizes a L_m-trade-off between the bias and the variance terms.

We consider the selection procedure based on a penalized criterion of the following form

$$\hat{m} = \arg\min_{m \in \mathcal{M}_n} [\gamma(\hat{f}_m) + \frac{\text{pen}_n(m)}{n}]$$

where $\text{pen}_n(m)$ is a penalty function mapping $\mathcal{M}_n$ into $\mathbf{R}^+$. We will precise this penalty later on and just mention that it is closely related to the classical C_p criterion of Mallows (1973).

The procedure is then the following: for each model m we compute the normalized residual sum of squares, $\gamma(\hat{f}_m)$, where γ is defined by (4), we choose $\hat{m}$ in order to minimize among all models $m \in \mathcal{M}_n$ the penalized residual sum of squares $\gamma(\hat{f}_m) + \text{pen}_n(m)/n$ and we compute the resulting estimator, $\hat{f}_{\hat{m}}$. Mallows' C_p criterion corresponds to $\text{pen}_n(m) = 2\hat{\sigma}^2 \dim(S_m)/n$ where $\hat{\sigma}^2$ denotes a suitable estimator of the unknown variance of the ε_i's. Our penalty term is similar but with an unknown universal constant instead of 2 and the factor L_m allowing for very rich collections of models (see the further discussion on the choice of the L_m's). When σ^2 is unknown we also replace it by an estimator.

2.2. Theoretical results

2.2.1. An example of theorem

From the theoretical point of view, Baraud (2000), Baraud (1998) and Baraud et al. (2001a,b) obtained several results depending mainly on the assumptions set on the error terms ε and on the types of the variables X_i in a more general model $Y_i = f(X_i) + \sigma\varepsilon_i$ where the X_i's can be random or deterministic, independent or mixing, independent of the ε_i's or not. We formulate in detail the result corresponding to the following condition:

($\mathbf{H}_\epsilon$) The ε_i's are i.i.d.centered variables and satisfy, $\forall u \in \mathbf{R}$

$$\mathbf{E}(\exp u\varepsilon_1) \leq \exp(u^2 s^2/2)$$

for some positive s.

This assumption allows the variables ε_i's to be Gaussian with variance s^2 or to be bounded by s. The particular case of Gaussian variables is given in Baraud (1998), and the following result is a simplified version of Theorem 2.1 in Baraud (1998) or Theorem 1 in Baraud et al. (2001b).

THEOREM 1. *Consider model (1) where f is an unknown function belonging to $\mathbf{L}_2([0,1])$. Assume that the ε_i's satisfy Assumption $(\mathbf{H}_\varepsilon)$ and that the family of piecewise polynomials described in section 2.1 has dimensions D_m such that*

$$\sum_{m \in \mathcal{M}_n} e^{-L_m D_m} \leq \Sigma < +\infty \tag{6}$$

where the L_m's are nonnegative numbers (to be chosen). Then there exists a universal constant $\theta > 0$ such that if the penalty function is chosen to satisfy

$$\mathrm{pen}_n(m) \geq \theta s^2 D_m(1 + L_m)$$

then the estimator $\tilde{f} = \hat{f}_{\hat{m}}$ satisfies

$$d_n^2(f, \tilde{f}) \leq C \inf_{m \in \mathcal{M}_n} \left[d_n^2(f, S_m) + \frac{\mathrm{pen}_n(m)}{n} \right] + C' s^2 \frac{\Sigma}{n} \tag{7}$$

where C and C' are universal constants.

Note that as $C' s^2 \Sigma / n$ has a smaller order that $\mathrm{pen}_n(m)/n$, the result in (7) may also be written:

$$d_n^2(f, \tilde{f}) \leq C(s^2, \Sigma) \inf_{m \in \mathcal{M}_n} \left[d_n^2(f, S_m) + \frac{\mathrm{pen}_n(m)}{n} \right]$$

where $C(s^2, \Sigma)$ denotes now a constant depending on s^2 and Σ.
This kind of result can be extended to variables ε_i's admitting only moments of order p, provided that $p > 2$ (see Baraud (2000)) for regular collections of models only. We shall try to see empirically if some problems arise in practice if this condition is not fulfilled by considering Cauchy ε's.
Note also that in many theoretical results, the multiplicative factor appearing in the penalty and here denoted by s^2 is in fact σ^2, i.e. the variance of the noise. If this variance is known, we keep it as the multiplicative factor. Else it can be estimated by the least square residuals:

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{f}_m(x_i))^2$$

for a $\hat{f}_m$ computed on a well chosen S_m. For instance, for equispaced design regression, we can take the space generated by $a_d = d/D$ with $D = n/\ln(n)$. This has been proved to allow an extension of the theoretical results in the case of regular subdivisions in Comte and Rozenholc (2002).

2.2.2. *Collections of models and choice of the weights*

Let us now illustrate condition (6) in order to better see the role of the L_m's. Roughly speaking, when the L_m's can be chosen constant, the final rate for estimating a function of smoothness α is the minimax rate $n^{-2\alpha/(2\alpha+1)}$. In most other cases, the L_m's are required to be of order $\ln(n)$ and the rate falls to $(n/\ln(n))^{-2\alpha/(2\alpha+1)}$. Let us give some (standard) examples for the choice of the spaces when the design is equispaced namely when $x_i = i/n$:

(RP) Regular piecewise polynomials (and regular S_m's). This is typically what is meant when talking about *regular* collections of models.
We work with constant degrees $r_1 = \ldots = r_D = r - 1$ and we choose $a_j = j/D$ for $j = 0, \ldots, D$ (regular partition of $[0, 1]$). Then $m = (D, a_1, \ldots, a_{D-1}, r, \ldots, r)$,

$\dim(S_m) = rD$, we take $D = 1, \ldots, D_{max}$ and we impose simply that $rD_{max} \leq n$, i.e. $D_{max} = [n/r]$. Then we look for L_m's such that

$$\sum_{m \in \mathcal{M}_n} e^{-L_m D_m} = \sum_{D=1}^{[n/r]} e^{-L_m D} \leq \Sigma < \infty.$$

Therefore

$$L_m = 1 \quad \text{or} \quad L_m = 2\ln(D)/D$$

suits.

(IPC) Irregular piecewise polynomials with constant degrees. This illustrates by comparison the extension from regular to *general* collections of models.
Once again we keep all the degrees constant equal to $r-1$. We choose the $D-1$ values of $a_1 < \ldots < a_{D-1}$ in the set $\{j/n, j \in \{1, \ldots, n-1\}\}$ for $D = 1, \ldots, D_{max} = [n/r]$. We have then for $L_m = L_n$

$$\sum_{m \in \mathcal{M}_n} e^{-L_m D_m} \quad = \quad \sum_{D=1}^{[n/r]} \left(\begin{array}{c} n-1 \\ D-1 \end{array} \right) e^{-rDL_n}.$$

Therefore, if we choose $L_m = L_n = \ln(n)/r$ this implies

$$\begin{aligned} \sum_{m \in \mathcal{M}_n} e^{-L_m D_m} &\leq \sum_{k=0}^{n-1} \left(\begin{array}{c} n-1 \\ k \end{array} \right) e^{-r(k+1)L_n} \\ &\leq \sum_{k=0}^{n-1} \left(\begin{array}{c} n-1 \\ k \end{array} \right) \left(\frac{1}{n}\right)^{k+1} = \frac{1}{n}\left(1 + \frac{1}{n}\right)^{n-1} \\ &\leq \left(1 + \frac{1}{n}\right)^{n} \leq e \end{aligned}$$

and condition (6) is satisfied.

(ITC) Irregular trigonometric polynomials with constant degrees.
The partitions and the a_j's are chosen previously. The degree in this example (but not in practice) is fixed also to $2r+1$ in the sense that, on an interval I of length ℓ we consider $\mathrm{Trig}_0^\ell(x) = \sqrt{1/\ell}\mathbb{1}_I(x)$,

$$\left\{ \begin{array}{l} \mathrm{Trig}_{2p}^\ell(x) = \sqrt{2/\ell}\cos\left(\frac{2n\pi}{\ell}px\right)\mathbb{1}_I(x), \\ \mathrm{Trig}_{2p+1}^\ell(x) = \sqrt{2/\ell}\sin\left(\frac{2n\pi}{\ell}px\right)\mathbb{1}_I(x), \end{array} \right.$$

for $p = 1, \ldots, r$. Let us mention that the Trig polynomials would have to be multiplicated by $\sqrt{n}$ to be normalized in $\mathbf{L}^2$. For the same reason as above, this would lead to weights L_m's of order $\ln(n)$, in order (6) to be fulfilled.

Note that the first example is meaningless for a non equally spaced design, but the second and third ones can be extended to the general fixed design case by simply choosing the knots $a_1 < \ldots < a_{D-1}$ in the set $\{x_1, x_2, \ldots, x_n\}$ with still $a_0 = 0$ and $a_D = 1$.

Moreover the degrees of the polynomials are supposed to be fixed (to $2r+1$) in the previous examples for the sake of simplicity but will be variable in the set $\{0,\dots,r_{max}\}$ in the algorithm developed below. In such case, the dimensional constraint $D_{max}=[n/r]$ becomes $\sum_{d=1}^{D}(r_d+1)\leq n$ where $r_d\in\{0,\dots,r_{max}\}$ is the degree of the polynomial on I_d. This implies a greater number of models.

2.2.3. Adaptation to unknown smoothness

It is easy to derive from inequalities like (7), adaptation results with respect to the unknown smoothness of f. We recall quickly that a function f belongs to the Besov space $\mathcal{B}_{\alpha,l,\infty}([0,1])$ if it satisfies

$$|f|_{\alpha,l}=\sup_{y>0}y^{-\alpha}w_d(f,y)_l<+\infty,\quad d=[\alpha]+1,$$

where $w_d(f,y)_l$ denotes the modulus of smoothness. For a precise definition of those notions we refer to DeVore and Lorentz (1993), Chapter 2, Section 7, where it also proved that $\mathcal{B}_{\alpha,p,\infty}([0,1])\subset\mathcal{B}_{\alpha,2,\infty}([0,1])$ for $p\geq 2$. This justifies that we now restrict our attention to $\mathcal{B}_{\alpha,2,\infty}([0,1])$. Moreover, if follows from DeVore and Lorentz (1993) that, if f_m is the orthogonal projection of f on S_m chosen in the collection **(IPC)** and if f belongs to $\mathcal{B}_{\alpha,2,\infty}([0,1])$ with $\alpha<r+1$, then $\|f-f_m\|^2$ is less than $C(\alpha)|f|^2_{\alpha,2}D_m^{-2\alpha}$.

If moreover the law μ of the X_i's admits is such that:

($\mathbf{H}_\mu$) μ admits a density h_μ with respect to the Lebesgue measure such that there exists $h_0>0$ and $h_1>0$ with $\forall x\in[0,1], 0<h_0\leq h_\mu(x)\leq h_1\leq 1$,

then, roughly speaking, $d_n(f,f_m)$ can be replaced by $\|f-f_m\|^2_\mu\leq h_1\|f-f_m\|^2$.

Then it follows from Inequality (7) of Theorem 1 that, under the above assumptions and the assumptions of Theorem 1,

$$d_n^2(f,\tilde f)\leq C(h_1,\alpha,r,s^2)\inf_{m\in\mathcal{M}_n}\left(|f|^2_{\alpha,2}D_m^{-2\alpha}+\frac{\ln(n)D_m}{n}\right)=C(h_1,r,\alpha,|f|_{\alpha,2},s^2)\left(\frac{n}{\ln(n)}\right)^{-\frac{2\alpha}{2\alpha+1}}.$$

This rate is reached by the estimator without requiring any prior information on α and this is what is called adaptation. Due to the large size of the collection of models, the rate reached is slightly sub-optimal (the optimal rate is known to be $n^{-2\alpha/(2\alpha+1)}$). A special strategy for visiting the spaces S_m is given in Proposition 4.1 and 4.2 of Baraud et al. (2001b) in order to recover the optimal rate even when considering non regular approximation spaces, but it is only a theoretical procedure for the moment. Here we can summarize our result as follows (see also Proposition 2.2 in Baraud (1998)):

COROLLARY 2. *Consider model (1) where f is an unknown function belonging to $\mathcal{B}_{\alpha,2,\infty}([0,1])$. Assume that the ε_i's satisfy Assumption* **($\mathbf{H}_\varepsilon$)**, *that the X_i's satisfy* **($\mathbf{H}_\mu$)** *and $\tilde f$ is computed in the collection* **(IPC)** *with $r+1>\alpha>0$ Then there exists a universal constant $\theta>0$ such that if the penalty function is chosen to satisfy*

$$\mathrm{pen}_n(m)\geq\theta s^2D_m(1+\ln(n)/r)$$

then the estimator $\tilde{f} = \hat{f}_{\hat{m}}$ *satisfies*

$$\sup_{f \in \mathcal{B}_{\alpha,2,\infty}([0,1]), |f|_{\alpha,2} \leq R} d_n^2(f, \tilde{f}) \leq C(h_1, r, \alpha, R, s^2) \left(\frac{n}{\ln(n)} \right)^{-\frac{2\alpha}{2\alpha+1}}$$

where $C(h_1, r, \alpha, R, s^2)$ *is a constant depending on* h_1, r, α, R, s^2.

For some results in the fixed design case and when considering regular spaces (see **(RP)**), we refer also to Proposition 4.1 in Baraud (2000): the optimal rate is also obtained if moreover the design is regular i.e. $x_i = i/n$ for $i = 1, \ldots, n$, if the ε's admit moments of order p with $p > 4$ and if $\alpha < r + 1$. For a general fixed design $x_1 < x_2 < \ldots < x_n$, the same rate is obtained asymptotically provided that the empirical measure associated to the design converges to a measure μ on $[0, 1]$ such that μ admits a density h_μ with: $\forall x \in [0, 1]$, $0 < h_0 \leq h_\mu(x) \leq h_1 \leq 1$. In other words, the limiting measure is required to be equivalent to the Lebesgue measure.

2.3. The aim of the calibration study

The order of the penalty as given in the theoretical results above is only a crude approximation that technically works and one of the aims of the empirical work is precisely to find a more precise development for the choice of the penalty. We also want to calibrate empirically some universal constants involved. For instance, if we think of a penalty:

$$\text{pen}_n(m) = s^2 \left[c_1 \ln \left(\begin{array}{c} n-1 \\ D-1 \end{array} \right) + c_2 (\ln(D))^{c_3} + c_4 \sum_{d=1}^{D} (r_d + 1) + c_5 \sum_{d=1}^{D} [\ln(r_d + 1)]^{c_6} \right] \quad (8)$$

we need to check that it satisfies (6). (Remind that m is defined by (2).) Then we want to prove empirically that the constants c_i, $i = 1, \ldots, 6$ are universal constants and compute them. Note that complementary terms in a penalty function have been studied in a theoretical framework (but for another problem and with a penalty having a different form) by Castellan (2000). On the other hand, empirical experiments for calibrating a penalty have already been lead for density estimation with regular histograms by Birgé and Rozenholc (2002). For all degrees set to zero and regular partitions, they proposed $\text{pen}_n(D) = [\ln(D)]^{2.5} + D - 1$. Here, we take $c_1 = c_4 = 2$ and $s^2 = \sigma^2$. We look for c_2, c_3, c_5 and c_6.

3. Computation of the estimate

3.1. The general formula

The first step for the computation of $\tilde{f} = \hat{f}_{\hat{m}}$ is the computation of the $\hat{f}_m$'s for m varying in $\mathcal{M}_n$ among which we choose it. Let $m = (D, a_1, \ldots, a_{D-1}, r_1, \ldots, r_D)$ be given and recall that $I_d = [a_{d-1}, a_d[$ for $d = 1, \ldots, D-1$, and $I_D = [a_{D-1}, a_D]$, $a_0 = 0$ and $a_D = 1$. Then $\hat{f}_m$ satisfies

$$\gamma(\hat{f}_m) = \frac{1}{n} \sum_{d=1}^{D} \min_{P} \sum_{x_k \in I_d} (Y_k - P(x_k))^2.$$

In other words, for some given m, we replace the global minimization of the contrast γ in S_m by D minimizations of local contrasts denoted

$$\gamma^{I_d}(g) = \frac{1}{n} \sum_{\{k/x_k \in I_d\}} (Y_k - g(x_k))^2 \tag{9}$$

on $\mathcal{P}_{r_d}$. In the algorithm, $\mathcal{P}_r$ will be either the linear space $\mathbf{R}_r[X]$ of standard polynomials of degree less or equal than r (see (**IPC**)) or the linear space $\mathbf{T}_r$ of trigonometric polynomials generated by the Trig_s^ℓ for $s = 0, \ldots, r$ (see (**ITC**)). Then we have to compute for any degree r and any interval I, the polynomial $P_r^I \in \mathcal{P}_r$ such that:

$$\hat{\gamma}_I(r) \stackrel{\text{def}}{=} \gamma^I(P_r^I) = \min_{P \in \mathcal{P}_r} \sum_{\{k/x_k \in I\}} (Y_k - P(x_k))^2/n = \frac{1}{n} \left[\sum_{\{k/x_k \in I\}} Y_k^2 - \sum_{\{k/x_k \in I\}} (P_r^I(x_k))^2 \right].$$

Note that this contrast is defined only by the points x_k and Y_k for the indexes k such that $x_k \in I$ and thus all intervals I' containing the same x_k lead to the same minimization procedure and to the same polynomial $P_{I'} = P_I$. So there is no loss of generality to consider intervals with bounds chosen among the x_k's.

It is well known that, for any base $\mathcal{B} := (B_0, \ldots, B_r)$ of a linear space $\mathcal{P}_r$, the contrast minimizer $P_r^I = \alpha_0 B_0 + \alpha_1 B_1 + \ldots + \alpha_r B_r$ is the solution of the system of equations $C_r^I A_r^I = D_r^I$ where (denoting by X' the transpose of the vector X),

$$A_r^I = (\alpha_0, \ldots, \alpha_r)', \tag{10}$$

$$C_r^I = (c_{s,t})_{1 \le s,t \le r}, \quad c_{s,t} = \sum_{k/x_k \in I} B_s(x_k) B_t(x_k), \tag{11}$$

$$D_r^I = (d_0, d_1, \ldots, d_r), d_s = \sum_{k/x_k \in I} Y_k B_s(x_k). \tag{12}$$

Let us denote by $\mathbf{X}_r^I$ the matrix $(B_s(x_k))$, $s = 0, \ldots, r$, $k \in \{j/x_j \in I\}$ with $r+1$ rows and with $\#\{k/x_k \in I\}$ columns, and by Y^I the vector of the Y_k's for x_k falling in I. The minimum of contrast satisfies

$$n\,\hat{\gamma}_I^{\mathcal{B}}(r) := n\,\hat{\gamma}_I(r) = (Y^I)'Y^I - (A_r^I)'\mathbf{X}_r^I(\mathbf{X}_r^I)'A_r^I = (Y^I)'Y^I - (D_r^I)'(C_r^I)^{-1}C_r^I(C_r^I)^{-1}D_r^I.$$

and therefore

$$\hat{\gamma}_I^{\mathcal{B}}(r) = \frac{1}{n}[(Y^I)'Y^I - (D_r^I)'(C_r^I)^{-1}D_r^I]. \tag{13}$$

3.2. Choice of a relevant base.

Since (13) with (10) is valid for any base $\mathcal{B}$ of $\mathcal{P}_r$, we look for a relevant choice of the base $\mathcal{B} = (B_0, \ldots, B_r)$ on the interval I, in term of piecewise polynomials. In other words, we aim at choosing the basis such that $C_r^I = I_r$ (I_r is the $r \times r$ identity matrix), that is

$$c_{s,t} = \sum_{k/x_k \in I} B_s(x_k) B_t(x_k) = \delta_{s,t}$$

where $\delta_{s,t}$ is the Kronecker symbol such that $\delta_{s,t} = 1$ is $s = t$ and 0 otherwise.

In the case of a general design $(x_i)_{1\leq i\leq n}$ (non necessarily equispaced), for each interval I, we can easily build by Gram-Schmidt orthonormalization or by using a Q-R decomposition of $\mathbf{X}$, an orthonormal basis of polynomials (whether standard or trigonometric) of any degree r, with respect to the discrete scalar product associated to the x_k's in I. The problem here is that for each possible interval I, and degree r_{max} a specific orthonormalized basis must be computed, which is feasible but quite heavy (and therefore quite slow) from a computational point of view. Consequently, some other ideas for accelerating the method have to be found.

3.3. Choice of a relevant base in the particular case $x_i = i/n$.

3.3.1. Polynomial base.

We use the discrete Chebyshev polynomials defined as follows (see Abramowitz and Stegun (1972)). The discrete Chebyshev polynomial on $\{0, 1, \ldots, \ell - 1\}$ with degree r is

$$\mathrm{Cheb}_r^\ell(x) = \frac{1}{\left\{\sum_{i=0}^{\ell-1}[C_r^\ell(i)]^2\right\}^{1/2}} C_r^\ell(x) \tag{14}$$

where $C_0^\ell(x) = 1$ and

$$C_r^\ell(x) = \frac{1}{(r!)^2}\Delta^r\left[\prod_{s=0}^{r} g_\ell(x-s)\right], \text{ where } g_\ell(x) = x(x-\ell)$$

and $\Delta f(x) = f(x+1) - f(x)$. Those polynomials satisfy

$$\sum_{k=0}^{\ell-1} \mathrm{Cheb}_r^\ell(k)\mathrm{Cheb}_s^\ell(k) = \delta_{s,t}, \text{ for } 0 \leq s, t \leq r.$$

Therefore, choosing on the intervals $I = [i/n, \ldots, (i+\ell+1)/n[$, the basis

$$B_s^I(x) = \mathrm{Cheb}_s^\ell(nx - i),$$

will do the job. This leads to

$$\hat{\gamma}_I^{\mathrm{Cheb}}(r) = \frac{1}{n}\left[(Y^I)'Y^I - (D_r^I)'D_r^I\right],$$

where D_r^I is the vector with components

$$d_s = \sum_{k/n\in I} Y_k B_s^I(k/n) = \sum_{k=0}^{\ell(I)-1} Y_{k+i}\mathrm{Cheb}_s^\ell(k).$$

3.3.2. Trigonometric base.

The case of piecewise trigonometric bases is even simpler since the basis described in **(ITC)** is naturally orthonormal with respect to the discrete scalar product considered with a regular design:

$$\sum_{x_k\in I} \mathrm{Trig}_s^\ell(x_k)\mathrm{Trig}_t^\ell(x_k) = \delta_{s,t}.$$

3.4. A fast strategy for the general case

We present here a way to avoid the Gram Schmidt orthonormalization for general design using the bases of the regular case given above.

Let us recall here that we assumed that all the x_i's were distinct and $0 \leq x_1 < x_2 < \ldots < x_n \leq 1$. Therefore the map Ψ that associates the normalized index i/n to x_i is a bijection, that can be defined from $[x_1, x_n]$ into $[1/n, 1]$ by setting for $x \in [x_i, x_{i+1}[$ that

$$\Psi(x) = \frac{i}{n} + \frac{x - x_i}{n(x_{i+1} - x_i)}. \tag{15}$$

Then we can solve the problem of looking for the piecewise polynomial $\hat{g}$ minimizing $\sum_{k=1}^{n}(Y_k - g(k/n))^2$ as previously with on each piece either the discrete Chebyshev base or the trigonometric base. Consequently we can consider the solution of the problem given by $\hat{f} = \hat{g} \circ \Psi$. This amounts to solve a regular problem and to distort the solution according to the design. An objection can be raised here. Indeed the procedure chooses an estimate $\tilde{g}$ such that $\tilde{f} = \tilde{g} \circ \Psi$ is an estimate of f. This implies, since

$$\begin{aligned} \|f - \tilde{f}\|_n^2 &= \frac{1}{n}\sum_{i=1}^{n}(\tilde{f} - f)^2(x_i) = \frac{1}{n}\sum_{i=1}^{n}(\tilde{g} \circ \Psi - f)^2(x_i) \\ &= \frac{1}{n}\sum_{i=1}^{n}\left[(\tilde{g} - f \circ \Psi^{-1}) \circ \Psi(x_i)\right]^2 = \frac{1}{n}\sum_{i=1}^{n}\left[(\tilde{g} - f \circ \Psi^{-1})(\frac{i}{n})\right]^2, \end{aligned}$$

that the rate of convergence of the estimate depends on the regularity of the function which is indeed estimated, namely $f \circ \Psi^{-1}$. Therefore, if we want to keep good theoretical rates of convergence for regular functions f, we need to use a function Ψ such that Ψ is increasing and Ψ^{-1} is very regular, satisfying $\Psi(x_k) = k/n$. This can be done by considering for instance, for $x \in [k/n, (k+1)/n[$,

$$\Psi^{-1}(x) = x_k + (x_{k+1} - x_k)\exp\left(1 - \frac{1}{n(x - k/n)}\right)\left[1 - \exp\left(-\frac{1}{(k+1)/n - x}\right)\right], \tag{16}$$

which is increasing an infinitely differentiable (with derivatives of any orders null at the points x_k). This has no practical impact since we look only at the discrete $\mathbf{L}_2$ norm and

$$\|f - \tilde{f}\|_n^2 = \frac{1}{n}\sum_{i=1}^{n}\left[\tilde{g}(i/n) - f(x_i)\right]^2,$$

which does not depend on Ψ.

3.5. The choice of the penalty

We already announced in equation (8) our choice for the global form of the penalty. For the results given in Theorem 1 to hold, we must prove that $\sum_{m \in \mathcal{M}_n} e^{-L_m D_m} < +\infty$ with $\mathrm{pen}_n(m) = s^2(1 + L_m)D_m$.

$$\sum_{m \in \mathcal{M}_n} e^{-L_m D_m} = \sum_{m \in \mathcal{M}_n} e^{-\mathrm{pen}_n(m)/s^2 + D_m}$$

$$
\begin{aligned}
&= \sum_{1\le D\le D_{max},(D,a_1,\ldots,a_{D-1},r_1,\ldots r_{D-1})} \exp\left\{-\left[c_1\ln\left(\begin{array}{c} n-1 \\ D-1 \end{array}\right)+c_2[\ln(D)]^{c_3}\right.\right. \\
&\qquad\qquad \left.\left.+c_4\sum_{d=1}^{D}(r_d+1)+c_5\sum_{d=1}^{D}[\ln(r_d+1)]^{c_6}\right]+D\right\} \\
&= \sum_{D=1}^{D_{max}}\left(\begin{array}{c} n-1 \\ D-1 \end{array}\right) e^{-\left[c_1\ln\left(\begin{array}{c} n-1 \\ D-1 \end{array}\right)+c_2[\ln(D)]^{c_3}\right]+D} \\
&\qquad\qquad \times\left[\sum_{r_1=0}^{r_{max}+1}\cdots\sum_{r_D=0}^{r_{max}} e^{-c_4\sum_{d=1}^{D}(r_d+1)-c_5\sum_{d=1}^{D}[\ln(r_d+1)]^{c_6}}\right] \\
&= \sum_{D=1}^{D_{max}} e^{\ln\left(\begin{array}{c} n-1 \\ D-1 \end{array}\right)-\left[c_1\ln\left(\begin{array}{c} n-1 \\ D-1 \end{array}\right)+c_2[\ln(D)]^{c_3}\right]+D}\left(\sum_{r=0}^{r_{max}} e^{-c_4(r+1)-c_5[\ln(r_d+1)]^{c_6}}\right)^D.
\end{aligned}
$$

Therefore, if $c_1\ge 1$, $c_2\ge 0$ and $c_5\ge 0$, we can give the following bound

$$
\sum_{m\in\mathcal{M}_n} e^{-L_m D_m} \le \sum_{D=1}^{D_{max}} e^D e^{-c_4 D}\left(\frac{1-e^{-c_4(r_{max}+1)}}{1-e^{-c_4}}\right)^D \le \sum_{D=1}^{D_{max}}\left(\frac{e^{1-c_4}}{1-e^{-c_4}}\right)^D,
$$

and this last term is bounded provided that

$$
\left|\frac{e^{1-c_4}}{1-e^{-c_4}}\right|<1
$$

that is, if $c_4>\ln(1+e)\simeq 1.3133$. Thus in the general case, the chosen penalty is of the form:

PROPOSITION 1. *The following choice of the penalty:*

$$
\mathrm{pen}_n(m)=s^2\left[c_1\ln\left(\begin{array}{c} n-1 \\ D-1 \end{array}\right)+c_2(\ln(D))^{c_3}+c_4\sum_{d=1}^{D}(r_d+1)+c_5\sum_{d=1}^{D}[\ln(r_d+1)]^{c_6}\right]
$$

is such that $\sum_{m\in\mathcal{M}_n} e^{-L_m D_m}=\sum_{m\in\mathcal{M}_n} e^{-\mathrm{pen}_n(m)/s^2+D_m}$ *converges with exponential rate, provided that* $c_1\ge 1$, $c_2\ge 0$, $c_4\ge 1.32$ *and* $c_5\ge 0$.

Let us mention that on each subinterval, we choose between a usual and a trigonometric polynomials, so that, if we denote by $R=\left(r_{max}^{(1)}+1+r_{max}^{(2)}+1\right)$ where $r_{max}^{(1)}$ and $r_{max}^{(2)}$ are the maximal degrees of each polynomials, the total number of visited bases is asymptotically (for great values of n and fixed R) of order

$$
\sum_{D=1}^{n}\left(\begin{array}{c} n-1 \\ D-1 \end{array}\right)R^D=R\,(R+1)^{n-1}=O(R^n).
$$

4. Description of the algorithm

In the sequel, both for the description of the algorithm and for the empirical results, we consider only the regular design defined by $x_i = i/n$. The algorithm is easy to generalize to non equispaced design (and works very well in terms of errors performance), but it is much too slow to be seriously considered for the moment.

4.1. Localization

Let us emphasize here the two basic ideas of our procedure. The first one is based on a **localization** of the problem. With the results and notations of Section 3.1 and the subsections following, the global value of the contrast is $\gamma(\hat{f}_m) := \hat{\gamma}_m = \sum_{d=1}^{D} \hat{\gamma}_{r_d}^{I_d}$ and we look for

$$\hat{m} = \arg \min_{m \in \mathcal{M}_n} \left[\hat{\gamma}_m + \frac{\mathrm{pen}_n(m)}{n} \right]$$

where we must consider here that $\mathrm{pen}_n(m) := \mathrm{pen}_{n,c}(m)$ with $c = (c_1, c_2, c_3, c_4, c_5, c_6)$ and

$$\begin{aligned} \mathrm{pen}_{n,c}(m) &= \sigma^2 \left[c_1 \ln \begin{pmatrix} n-1 \\ D-1 \end{pmatrix} + c_2 \ln(D)^{c_3} \right] + \sum_{d=1}^{D} \sigma^2 \left[c_4(1+r_d) + c_5 \ln(1+r_d)^{c_6} \right] \\ &\stackrel{\mathrm{def}}{=} \mathrm{pen}_{n,c_1,c_2}(D) + \sum_{d=1}^{D} \mathrm{pen}_{n,c_4,c_5,c_6}(r_d). \end{aligned}$$

Then we find a localized decomposition of the penalized contrast:

$$n\hat{\gamma}_m + \mathrm{pen}_{n,c}(m) = \mathrm{pen}_{n,c_1,c_2}(D) + \sum_{d=1}^{D} \left\{ n\hat{\gamma}_{I_d}^{\mathcal{B}}(r_d) + \mathrm{pen}_{n,c_4,c_5,c_6}(r_d) \right\},$$

where the first part of the penalty $\mathrm{pen}_{n,c_1,c_2}(D)$ is the global penalization concerning the number of sub-intervals and the second part $\mathrm{pen}_{n,c_4,c_5,c_6}(r_d)$ is the local part concerning the degree on each sub-interval. We recall that $\hat{\gamma}_I^{\mathcal{B}}(r)$ is defined by (13) for a basis $\mathcal{B}$.

We implement in fact a multi-bases estimate. For this purpose, the algorithm chooses on each subinterval a particular base between a preselected collection of bases. Here our preselected collection of bases is, as described in section 3.3, the "Cheb" and "Trig" bases. For this multi-bases implementation the decomposition of the penalized contrast is as follows:

$$n\hat{\gamma}_m + \mathrm{pen}_{n,c}(m) = \mathrm{pen}_{n,c_1,c_2}(D) + \sum_{d=1}^{D} \min_{\varphi_d} \left\{ n\hat{\gamma}_{I_d}^{\varphi_d}(r_d) + \mathrm{pen}_{n,c_4,c_5,c_6}^{\varphi_d}(r_d) \right\}.$$

This can also be written:

$$n\hat{\gamma}_m + \mathrm{pen}_{n,c}(m) = \sum_{k=1}^{n} Y_k^2 + \mathrm{pen}_{n,c_1,c_2}(D) + \sum_{d=1}^{D} \left[\mathrm{pen}_{n,c_4,c_5,c_6}^{\varphi_d}(r_d) - p_{I_d}^{\varphi_d}(r_d) \right]$$

where on the interval $I = [i/n, \ldots, (j+1)/n[$

$$p_I^{\varphi}(s) = \sum_{t=0}^{s} \left[\sum_{k=0}^{j-i} Y_{k+i} \varphi_t^I(k) \right]^2 \stackrel{\text{def}}{=} p_s^{\varphi}(i,j),$$

where $\varphi^I = (\varphi_0^I, \ldots, \varphi_s^I, \ldots)$ is one of the preselected bases and φ_d is set for φ^{I_d}. The quantity $p_s^{\varphi}(i,j)$ represents obviously the weight of the contrast when going from i to j (j included), so that $p^{\varphi}(i,i)$ is defined. Note that, for $1 \leq \ell \leq n$, those quantities are systematically computed by setting

$$\mathbf{Y}_\ell = (Y_{i+k-1})_{1 \leq i \leq \ell, 1 \leq k \leq n-\ell} \text{ and } \mathbf{B}_\ell = (\varphi_i^I(k))_{0 \leq i \leq r_{max}, 0 \leq k \leq \ell-1}$$

and by computing and storing $(\mathbf{B}_\ell \mathbf{Y}_\ell)^{\bullet 2}$ where $A^{\bullet 2} = (a_{i,k}^2)_{1 \leq i \leq p, 1 \leq k \leq q}$ for $A = (a_{i,k})_{1 \leq i \leq p, 1 \leq k \leq q}$. Then considering different values of ℓ amounts to take into account intervals of any length $\ell = 1, \ldots, n$. Note that even if pen^{φ} can depend on φ we choose after some experiments the same penalization for each base.

For $j \geq i$, the procedure of minimization first computes:

$$p(i,j) = \min_{\varphi} \min_{1 \leq s \leq r_{max}} \left[\sigma^2 (c_4(1+s) + c_5 \ln(1+s)^{c_6}) - p_s^{\varphi}(i,j) \right],$$

so that the best base and the best degree is chosen.

4.2. Dynamical programming

We reach here the point where we need to use **dynamical programming** (see Kanazawa (1992)). The fundamental idea of dynamical programming here is that to go until point j with d steps (pieces here), we must first go until some $k < j$ with $d-1$ steps and then go from k to j in one step.

Let $q(d,k)$ be the minimum of the contrast − penalized in degree with base selection − to go from 1 to k with d pieces; this value is thus associated to a best partition, d best bases and a choice of d best degrees which fulfill the localization constraints.

First note that $q(1,k) = p(1,k)$ which gives an initialization; then

$$q(d+1,j) = \min_{d \leq k < j} \left[q(d,k) + p(k+1,j) \right] \tag{17}$$

which represents $2j$ operations. Then a Q matrix can be filled in, with two possible strategies:

(a) "Off line" method: Compute the $q(1,j)$ for $j = 1, \ldots, n$ and then do a recursion on d using (17). The drawback of the method is that the actualization (i.e. if some more observations are available and n changes), everything must be done again whereas you know that the last column only changes.

(b) "In line" method: Assume that you have built $(q(d,j))_{1 \leq d \leq j \leq n}$ and you want to increase n and compute the $q(d,n+1)$, $d = 1, \ldots, n+1$. Then as $q(1,n+1) = p(1,n+1)$ and

$$q(d+1,n+1) = \min_{d \leq k < n+1} \left[q(d,k) + p(k+1,n+1) \right],$$

you only need to compute the $p(k+1,n+1), 1 \leq k \leq n$, the $q(d,k)$ being already known.

The first part of the work, namely the computation of the coefficients $p(i,\ell)$ requires $O(n^3 r_{max})$ elementary operations, and the dynamical programming part requires $O(n^2 D_{max})$ operations. The global complexity of the algorithm is therefore of order

$$n^3 r_{max} + n^2 D_{max}.$$

The implemented method is the first one (Off line), but of course, for an actualization purpose, the second method must be preferred.

Now, on the last column of Q, there are the $q(d,n)$'s, $1 \leq d \leq n$, which are the minima of the contrast penalized in degree, to go from 1 to n with d pieces. Thus the last thing to do is to choose

$$\hat{D} = \arg \min_{d=1,\ldots,n} \left[q(d,n) + c_1 \ln \left(\begin{array}{c} n-1 \\ d-1 \end{array} \right) + c_2 \ln(d)^{c_3} \right].$$

Of course, the involved partitions must be stored, and not only their number of pieces.
As a summary, let us give the steps of the algorithm:

PROPOSITION 2. *A model is selected by the algorithm following the steps:*

1. *On any interval* $I = [i/n, (j+1)/n[$, *compute* $p_s^{\varphi}(i,j) = \sum_{t=0}^{s} \left[\sum_{k=0}^{j-i} Y_{k+i} \varphi_t^I(k) \right]^2$ *for* $1 \leq i \leq j \leq n$, $0 \leq s \leq r_{max}$, *and for* $\varphi_t^I = \mathrm{Cheb}_t^{\ell(I)}$ *and* $\varphi_t^I = \mathrm{Trig}_t^{\ell(I)}$ *(see Section 3.3.1 and 3.3.2),*

2. *Compute* $p^{\varphi}(i,j) = \min_{1 \leq s \leq r_{max}} \left(\sigma^2 (c_4 s + c_5 \ln(s)^{c_6}) - p_s^{\varphi}(i,j) \right)$ *for* $1 \leq i \leq j \leq n$,

3. *Compute* $p(i,j) = \min_{\varphi \in \{\mathrm{Cheb}, \mathrm{Trig}\}} p^{\varphi}(i,j)$,

4. *Initialize* $q(1,k) = p(1,k)$ *for* $1 \leq k \leq n$, *and compute recursively for* $1 \leq d \leq n-1$,

$$q(d+1,n) = \min_{d \leq k < n} \left[q(d,k) + p(k+1,n) \right],$$

5. *Then choose* $\hat{D} = \arg \min_{d=1,\ldots,n} \left[q(d,n) + c_1 \ln \left(\begin{array}{c} n-1 \\ d-1 \end{array} \right) + c_2 \ln(d)^{c_3} \right]$.

The positions of the knots of the involved partitions as well as the selected degrees in step 2 must be stored.

4.3. A fast version of the algorithm

We also implemented a quick but approximated version of the algorithm, with complexity of order $D_{opt} r_{max} n^2$.
We do not describe it with much details, but simply with its global idea. Namely, each step of the algorithm answers to the question: is it better to add one point to the subdivision or to cancel one, where the "better" is evaluated in term of the compared penalized contrasts. We must admit that many available algorithms are with complexity of order $O(n \log_2(n))$, this is really a drawback of our procedure.

5. Empirical results

5.1. Risks and calibration

First we take for the penalty as defined in Proposition 1, $s^2 = \sigma^2$, $c_1 = c_4 = 2$. We look for c_2, c_3, c_5 and c_6 with the constraint $(c_2, c_3) = (c_5, c_6)$. Indeed, the symmetry of those coefficients seems a natural property here and we need to limit the number of possibilities to explore.
Second, we use a set of 16 test functions with very different shapes and regularity. The test functions are given in Figure 1. Functions 1 to 4 and 7 to 14 are the same as the ones used by Antoniadis et al. (2002), functions 1 to 14 come from the Wavelab toolbox developed by Donoho (see Buckheit et al. (1995)) and functions 15 and 16 have been added in order to test also the estimators for some regular functions.

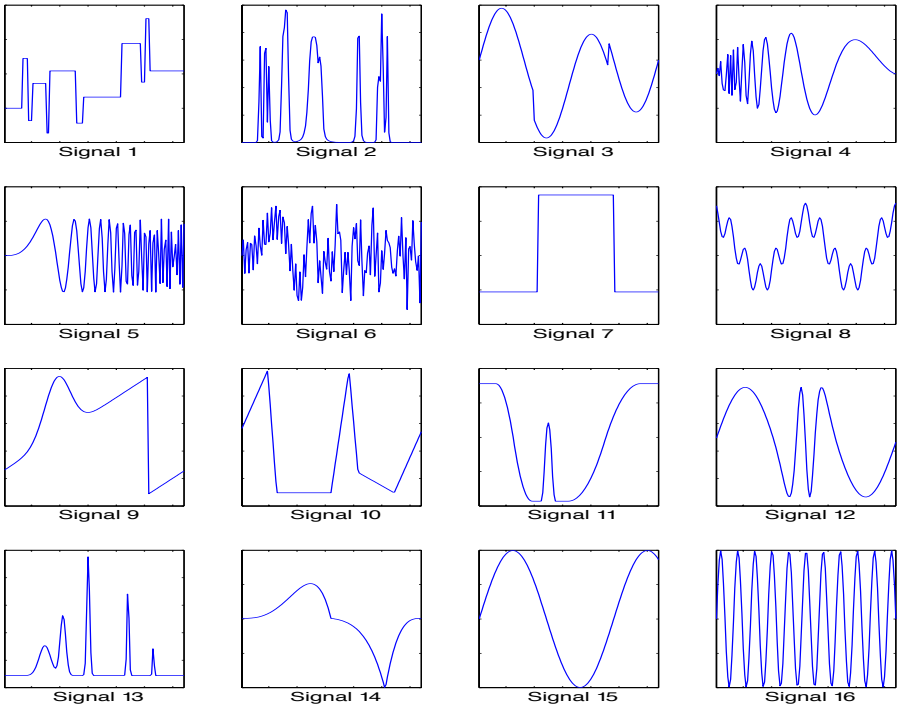

Fig. 1. Test functions.

Third, we consider different levels (namely 3, 5, 7 , 10) of noise which are evaluated in

terms of a signal to noise ratio, denoted by s2n, and computed as

$$\text{s2n} = \sqrt{\frac{\frac{1}{n}\sum_{i=1}^{n}(f(i/n)-\bar{f})^2}{\sigma^2}}, \quad \bar{f} = \frac{1}{n}\sum_{i=1}^{n} f(i/n).$$

Lastly, the performances are usually compared to a reference value called an oracle. This oracle is the lowest value of the risk. It is computed by using the fact that, in the simulation study, the true function is known, and that, if all the estimates are computed, the one with the smallest risk can be found, as well as its associated risk. In other words here, we would have to compute all $\|f-\hat{f}\|_n^2$, for some known f and all possible $\hat{f}$, for all the sample paths. This can be done for regular models but would imply much too heavy computations here for general models. Therefore, another reference must be found to evaluate our estimator. We use the following ratios

$$R_1(f) := \frac{\min_{j=1,\dots,48} \mathbf{E}^*\left[\ell_2^2(f,\hat{f}_{w_j})\right]}{\mathbf{E}^*\left[\ell_2^2(f,\tilde{f})\right]} \text{ and } R_2(f) = \frac{\mathbf{E}^*\left[\min_{j=1,\dots,48} \ell_2^2(f,\hat{f}_{w_j})\right]}{\mathbf{E}^*\left[\ell_2^2(f,\tilde{f})\right]} \tag{18}$$

the second one being of course a harder criterium than the first one to evaluate our method. In both cases, the ratios are compared to one: the higher over 1 the ratio, the better our method. The index w_j denotes the wavelet method number j where 48 wavelet methods are considered and $\hat{f}_{w_j}$ denotes the estimate of f obtained using the method w_j. Before giving the details about the wavelet methods, let us explain formula (18). We generate K ($K = 100$) samples with length n ($n = 128, 512$) in the regression model, and denote by $\hat{f}^{(k)}$an estimate of f (computed with any method, $\hat{f}^{(k)}_{w_j}$ with the method w_j) based on the k^{th} sample. Then

$$\ell_2^2(f,\hat{f}^{(k)}) = \frac{1}{n}\sum_{i=1}^{n}(f-\hat{f}^{(k)})^2(i/n),$$

and

$$\mathbf{E}^*[\ell_2^2(f,\hat{f})] = \frac{1}{K}\sum_{k=1}^{K}\ell_2^2(f,\hat{f}^{(k)}).$$

Therefore, if our test functions $f_1,\dots f_{16}$ lead to values of $R_2(f_i)$ for $i = 1,\dots,16$ such that $\forall i \in \{1,\dots,16\}$, $R_2(f_i) \geq a$, then this means that, for any $f \in \{f_1,\dots,f_{16}\}$,

$$\mathbf{E}^*[\ell_2^2(f,\tilde{f})] \leq \frac{1}{a}\mathbf{E}^*\left[\min_{j=1,\dots,48}\ell_2^2(f,\hat{f}_{w_j})\right].$$

We must emphasize that we chose diadic values of n ($n = 128 = 2^7$ or $n = 512 = 2^9$) in order to be able to apply all wavelet methods, but our method does not require diadic samples and can be used for any n without any change.

Now let us be more precise about the wavelets. We use both the MathWorks toolbox developed by Misiti et al. (1995) and the WaveLab toolbox developed by Buckheit et

al. (1995), following the theoretical works by Donoho (1995), Donoho and Johnstone (1994), Donoho et al. (1995). The abbreviations below refer to the MathWorks toolbox. We use the 6 following basic wavelets: the Haar wavelet (well suited for square signals), two Daubechies DB4 and DB15 wavelets (well suited for smooth signals), two symmetric wavelets abbreviated as Symmlets, sym2 and sym8, the bi-orthogonal wavelet bior3.1 (well suited for signals with rupture). The wavelets are associated with 4 types of threshold: the threshold $\sqrt{2\log(n)}$ called "sqtwolog", the minimax threshold called "minimaxi", the SURE (Stein's Unbiased Risk Estimate) threshold called "Rigsure", an heuristical version of SURE threshold using a correcting term for small values of n, called "Heursure". Lastly, we use the two standard types of threshold, hard and soft thresholding†. This explains the $6 * 4 * 2 = 48$ indexes for the wavelets methods.

Several attempts and experiments lead, for the set of constant of the penalty, to the choice

$$c_1 = c_2 = c_4 = c_5 = 2, c_3 = c_6 = 2.5.$$

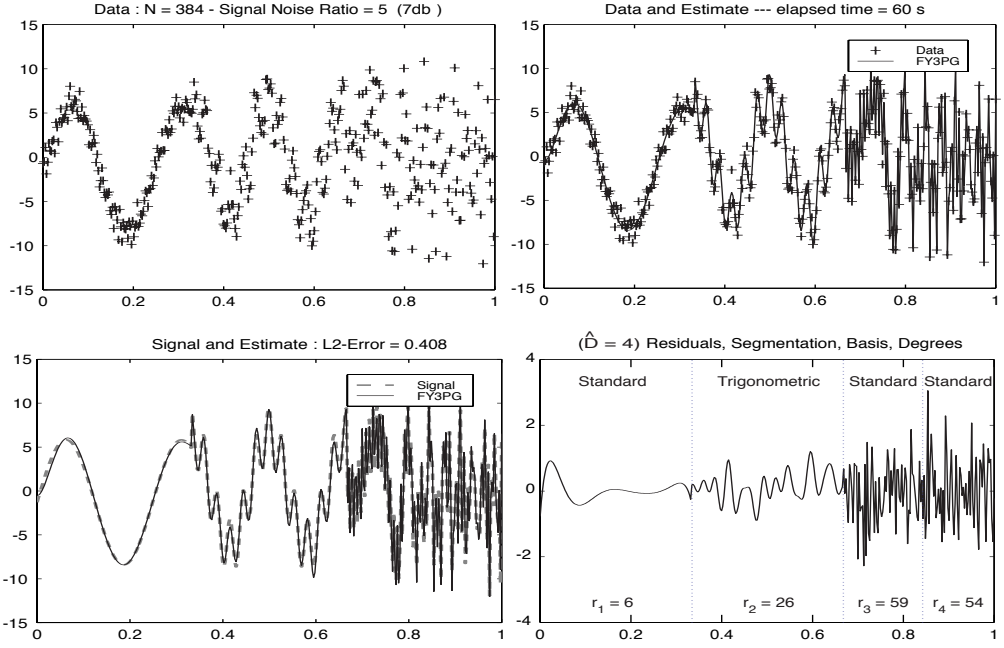

Fig. 2. An example of decomposition of a signal by the complete algorithm.

†Note that the estimation is improved by using the function "wmaxlev" to select the maximum level of the wavelets instead of the standard level round$[\log_2(n)]$ and we therefore use this MathWorks function as well.

We present in Figure 2 an example of data set and estimated signal as performed by our algorithm. The signal has been built with three pieces using functions 15, 8 and 6. The fourth picture gives the variation of the residuals and shows that the algorithm has found an estimator with four pieces, the first one is a standard polynomial with degree 6 corresponding to the estimation of the first function, the second one is a trigonometric polynomial with degree 26 corresponding to the estimation of the second function, the last two pieces correspond to the estimation of the third function, and are polynomials of degree 59 and 54.

5.2. Comparison with standard wavelet methods

We report more systematically in Figure 3 the performances of this choice when the maximal degree is set to $r_{max} = 74$ and for s2n$= 3, 5, 7, 10$, the functions f_i being as given in Figure 1, $K = 100$ and $n = 128$.

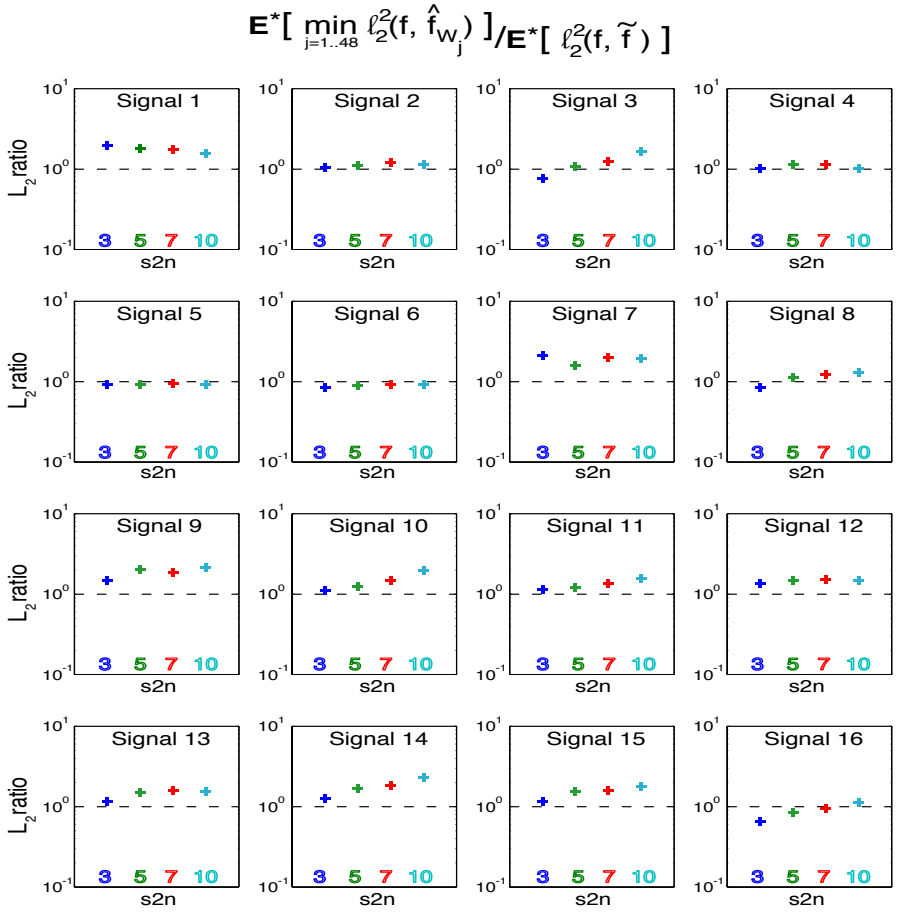

Fig. 3. Performance ratios $R_2(f)$ relatively to the 16 test functions for $K = 100$ and $n = 128$. The greater than one, the better our algorithm.

More precisely, Figure 3 plots the values obtained for $R_2(f)$ relatively to the test functions given in Figure 1. We emphasize that the ratio we compute is very unfavourable to our method, because for each sample, we compare our risk to the one of the best (unknown in practice) wavelet method. The ordinate of the lower point is anyway greater than 0.65, which is quite good since we recall that it means that, for any $f \in \{f_1, \dots, f_{16}\}$, $\mathbf{E}^*[\ell_2^2(f, \tilde{f})] \leq 1.54\mathbf{E}^* \left[\min_{j=1,\dots,48} \ell_2^2(f, \hat{f}_{w_j})\right]$.

Note that we have also computed the risks in term of ℓ_1-type error; we obtain the same type of results except for the function f_8 and f_{16} where the results are even better: this can be explained by the fact that the ℓ_1-distance reduces the weights of the discontinuities which are inherent in our method. Besides, the errors are taken centered Gaussian with known variance, but we also considered centered uniform and Cauchy errors and the results were similar.

We have roughly described in Section 4.3 an accelerated version of our complete algorithm, but we had to test if the performances of this method were indeed of the same order as the standard one, but appreciably faster. We give below in Table 1 the estimation performances in term of $R_2(f)$ and in term of CPU time (for the same samples) of the accelerated algorithm compared to the standard one.

Signal \ Ratio	s2n = 3		s2n = 5		s2n = 7		s2n = 10	
	Risk	Time	Risk	Time	Risk	Time	Risk	Time
1	1.55	0.24	1.53	0.25	1.37	0.25	1.46	0.25
2	0.93	0.18	0.95	0.18	0.96	0.17	0.99	0.16
3	1.05	0.10	1.00	0.11	1.01	0.12	1.00	0.14
4	0.95	0.13	0.95	0.13	0.96	0.14	0.96	0.14
5	0.99	0.19	1.00	0.19	0.99	0.18	0.98	0.18
6	0.99	0.19	1.00	0.20	0.99	0.20	1.00	0.21
7	1.00	0.13	1.00	0.14	1.00	0.14	1.00	0.14
8	0.97	0.08	0.98	0.08	0.98	0.08	1.00	0.08
9	0.96	0.11	0.96	0.13	0.96	0.13	0.95	0.13
10	0.93	0.13	1.04	0.14	1.04	0.15	1.21	0.15
11	1.03	0.10	1.11	0.12	1.10	0.13	1.03	0.14
12	1.00	0.13	1.04	0.13	1.07	0.13	1.09	0.13
13	0.93	0.17	0.97	0.17	0.98	0.16	1.01	0.16
14	0.95	0.10	0.94	0.10	0.92	0.10	0.94	0.10
15	0.96	0.05	0.98	0.05	1.00	0.06	0.99	0.07
16	1.04	0.13	1.04	0.13	1.14	0.13	1.08	0.13

Table 1. Quick and complete algorithm comparison : Risk ratio and CPU Time ratio, ratio = Quick / complete, $n = 512$ and $K = 100$.

It appears that except for the first signal, which is better identified by the complete algorithm, the quick algorithm performs very well both in term of risk (which was expected) and time (which was the aim). More precisely and if we except Signal 1, there is essentially no loss in term of risk when using the quick algorithm, but it appears indeed to be between five and twenty times faster for a sample with size $n = 512$. This effect naturally increases with the sample size. As a conclusion, it is clear that both the standard the accelerated algorithm work very well.

5.3. Compression performances

We already computed the complexity of our algorithm so that it is clear that even in its quick version, it remains slower than wavelet methods. But it has two decisive advantages with respect to those methods, in addition to its completely automatic feature: first, it performs very well whatever the type of signal, and even faced with discontinuities, and second, its compression properties are quite excellent, and in particular much better in many cases than wavelets, which was somehow unexpected.

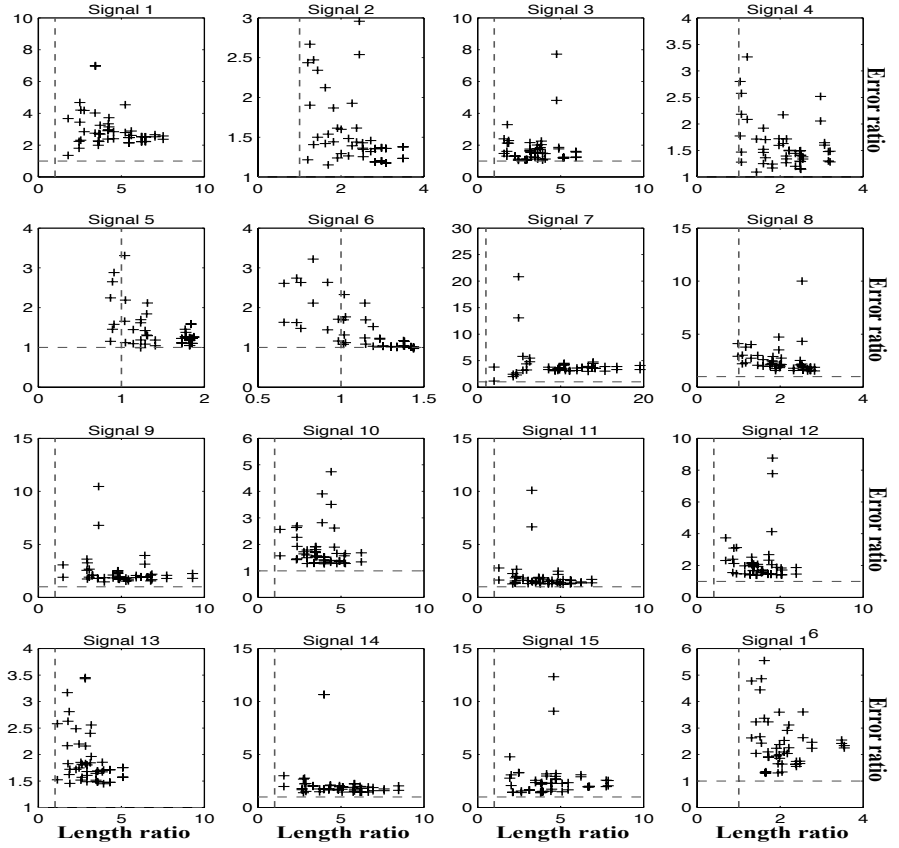

Fig. 4. Error ratio in function of the compression ratio. The ratios are our method over all wavelet methods (dotted lines correspond to levels 1). $K = 100$ samples with length $n = 128$ and s2n=5.

Therefore, we also lead a naive comparison of the standard wavelet methods and of our algorithm in term of their compression performances. For each estimated function, we compute three types of code lengthes: an "integer length" which is a number of integers, namely the number of nonzero wavelet coefficients for the wavelet methods, $NintW$, and twice the chosen number D of intervals in the piecewise polynomials method $NintPP$ (1

integer for D, D integers for each degree r_d, $D-1$ integers for the length of the intervals), a "real length" which is the number of real coefficients of the developments for each method, $NrealW$ and $NrealPP$, and a global length, NW and NPP, defined in both cases as $(Nint/4) + Nreal$ to take into account the common idea that an integer is four times smaller than a real number in terms of code length. Figure 4 above plots the error ratios $R^{(j)}(f) := \mathbf{E}^* \left[\ell_2^2(f, \hat{f}_{w_j})\right] / \mathbf{E}^* \left[\ell_2^2(f, \tilde{f})\right]$ in function of the global length ratios NW_j/NPP where j is the index of the wavelet method, for f taken as each signal of Figure 1 and for s2n= 5. We can see that both the estimation and the compression performances of our algorithm are most of the time better than wavelet methods since both ratios are higher than 1. The signal 5 and 6 are the only one for which the wavelets are better but it appears that they are better either in term of risk or in term of compression, but not both; on the contrary for the other signals, our method is better in term of both risk and compression performance.

We give also in Table 2 the means over the samples of the ratios: mean (over the samples) of $NrealW_j$ over mean of $NrealPP$, mean of $NintW_j$ over mean of $NintPP$ and mean of NW_j over mean of NPP, for the index j corresponding to the best wavelet in term of approximation performance for each sample path; we also distinguish between the values s2n of the signal to noise ratios. We can see that all ratios are greater than 1, which means that our method is better in term of compression that all standard wavelets. Besides, the compression improvement of our algorithm increases when the s2n increases.

	$NrealW_j/NrealPP$				$NintW_j/NintPP$				NW_j/NPP			
	s2n				s2n				s2n			
Signal	3	5	7	10	3	5	7	10	3	5	7	10
1	2,57	2,05	2,09	2,22	1,61	1,20	1,24	1,26	2,30	1,79	1,84	1,93
2	2,92	2,89	1,52	1,47	4,32	4,20	2,35	2,44	3,13	3,08	1,64	1,59
3	2,85	3,57	3,33	3,24	5,25	6,46	6,31	7,30	3,14	3,92	3,68	3,64
4	2,23	2,12	2,11	1,20	7,61	8,72	9,74	6,86	2,59	2,50	2,50	1,44
5	1,47	1,01	1,17	1,06	16,74	12,58	14,98	13,64	1,80	1,23	1,43	1,29
6	1,13	1,16	1,19	1,22	28,89	33,98	35,01	36,30	1,40	1,44	1,47	1,52
7	2,35	2,26	2,40	2,36	1,32	1,33	1,35	1,34	2,03	1,98	2,08	2,05
8	1,64	2,33	2,55	2,43	9,06	18,00	21,97	22,10	1,96	2,83	3,10	2,95
9	1,35	3,71	3,55	2,80	1,80	5,56	5,69	4,94	1,42	3,97	3,84	3,06
10	3,00	4,97	5,19	5,75	4,80	6,87	6,26	6,27	3,24	5,26	5,37	5,84
11	2,18	4,72	4,85	4,68	3,54	8,07	8,89	8,84	2,36	5,15	5,33	5,17
12	4,43	3,15	4,67	4,50	8,79	7,48	11,41	11,52	4,92	3,57	5,30	5,12
13	3,50	3,67	3,68	1,85	5,10	5,40	5,63	2,93	3,73	3,92	3,95	1,99
14	3,40	2,49	2,46	6,11	5,49	4,53	4,68	12,79	3,68	2,73	2,72	6,82
15	2,09	3,45	3,14	3,21	3,42	6,41	6,04	6,85	2,26	3,80	3,47	3,59
16	1,40	1,61	1,83	1,97	9,95	13,01	15,99	18,50	1,69	1,95	2,23	2,40

Table 2. Compression performances for the three ratios in function of the s2n and of the signal for $K = 100$ samples with length $n = 128$.

5.4. Comparison with recent wavelet methods

We also implemented for comparison some more recent wavelet methods, already studied in Antoniadis et al. (2002) and therefore quite reproducible for such a test. More precisely we considered the following methods, implemented using the Gaussian Wavelet Denoising Library built by Antoniadis et al. (2002) (see http://www.jstatsoft.org/v06/i06/), using either a Haar or a Symmlet8 filter:

W1 Coifman and Donoho (1995)'s translation invariant method using soft thresholding (TI-soft), coded with the function "recTI" in the library,
W2 Coifman and Donoho (1995)'s translation invariant method using hard thresholding (TI-hard), coded with the function "recTI" in the library,
W3 Cai (1999)'s method using a block non-overlapping thresholding estimator, re-using the first few empirical coefficients to fill the last block, coded with the function "recblockJS" in the library
W4 Cai (1999)'s previous method, the last few remaining empirical coefficients being unused, coded with the function "recblockJS" in the library,
W5 Huang and Lu (2000)'s method based on nonparametric mixed-effect models, coded with the function "recmixed" in the library,
W6 Cai and Silverman (2001)'s method using an overlapping block thresholding estimator, coded with the function "recneighblock" in the library,
W7 Antoniadis and Fan (2001)'s hybrid method using a "keep", "shrink" or "kill" rule (SCAD),
W8 Vidakovik and Ruggeri (2000)'s bayesian adaptive multiresolution method coded with the function "recbams" in the library.

Signal	PP	PP/PT	W1	W2	W3	W4	W5	W6	W7	W8
1	**0.077**	**0.098***	1.856	0.423$_\bullet$	0.798	0.817	0.472	0.947	0.866	0.871
2	**0.313**	**0.361***	2.421	0.404$_\bullet$	0.769	0.785	0.506	0.820	0.977	0.871
3	**0.061**	**0.063***	0.207	0.108$_\bullet$	0.198	0.228	0.135	0.216	0.148	0.856
4	**0.197**	**0.202**	0.805	0.174$^*_\bullet$	0.251	0.282	0.238	0.197	0.335	0.857
5	**0.582**	**0.563***	4.337	0.651$_\bullet$	0.724	0.746	0.670	0.677	1.600	0.883
6	**1.001**	**1.003***	8.506	1.601	1.492	1.474	4.788	1.406$_\bullet$	3.816	4.050
7	**0.010**	**0.012***	0.552	0.133$_\bullet$	0.388	0.405	0.193	0.416	0.285	0.860
8	**0.168**	**0.059***	0.596	0.072$_\bullet$	0.239	0.256	0.273	0.253	0.398	0.864
9	**0.050**	**0.053***	0.377	0.080$_\bullet$	0.159	0.177	0.143	0.198	0.159	0.859
10	**0.113**	**0.091**	0.313	0.084$^*_\bullet$	0.127	0.154	0.120	0.135	0.181	0.859
11	**0.087**	**0.087**	0.200	0.052$^*_\bullet$	0.108	0.140	0.088	0.082	0.091	0.856
12	**0.084**	**0.083**	0.235	0.050$^*_\bullet$	0.085	0.117	0.094	0.080	0.106	0.858
13	**0.181**	**0.141**	0.891	0.136$_\bullet$	0.258	0.269	0.234	0.246	0.365	0.867
14	**0.057**	**0.052**	0.176	0.062$^*_\bullet$	0.097	0.096	0.073	0.096	0.084	0.857
15	**0.027**	**0.027***	0.203	0.071$_\bullet$	0.156	0.175	0.136	0.166	0.122	0.856
16	**0.156**	**0.076***	0.735	0.118$_\bullet$	0.233	0.251	0.255	0.227	0.357	0.868

Table 3. $\mathbf{L}_2$-errors for s2n= 5, $n = 512$, PP is our method when considering piecewise polynomial bases only, PP/PT is our method when considering both standard and trigonometric piecewise polynomials, W1 to W8 are the wavelet methods described above with Symmlet8 filter. $\sigma = 1$ is known. $_\bullet$ gives the best wavelet method, * gives the best method between PP/TT and W1-W8.

For a more precise description of those methods, we refer to Antoniadis et al. (2002). They are a selection of recent methods that Antoniadis et al. (2002) describe and test, namely methods number 5, 6, 12, 13, 20, 11, 18 and 34 respectively in their Table 3. We work first with $\sigma = 1$ assumed to be known. Moreover, in all the following, we use the quick version of our algorithm.

Table 3 above gives the $\mathbf{L}_2$-errors for the 16 test functions and signal to noise ratio s2n=5 obtained with the quick version of our method (when using standard polynomials (PP) or both standard and trigonometric polynomials (PP/PT)) and with the other methods W1 to W8. We must say that we did not succeed in making W8 work, but this may be an error of ours. Besides we found out that the method of Coifman and Donoho (1995) with hard thresholding (W2 or TI-hard) seems to be almost always better than all the other wavelet methods. Our method behaves very well and is in general better than all the other methods. Even when we do not have the lowest errors, we are not far from it. Globally, the PP/PT method seems to be preferable: the losses are never very important but the gains are sometimes decisive, when compared to the wavelet methods in competition.

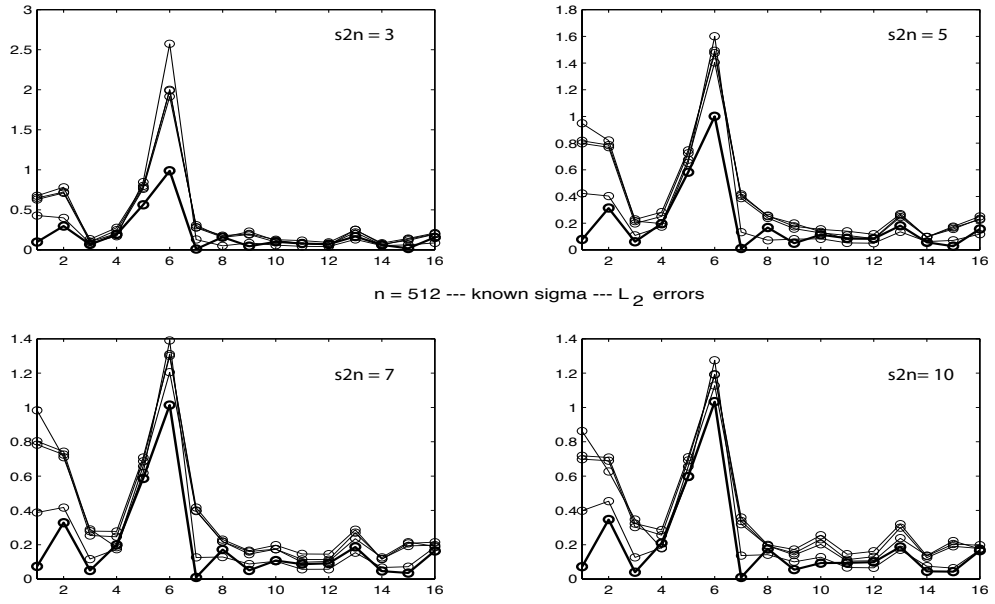

Fig. 5. Comparison of the $\mathbf{L}_2$-errors for the 16 test functions and the four s2n ratios. Our piecewise polynomial method is the thick curve, the other curves are the best wavelet methods among W1–W8 with Symmlet8 filter. $K = 100$, $n = 512$, $\sigma = 1$ known.

In order to give a better idea of the dependence with respect to the value of s2n and visualize the level of the performances, Figure 5 gives the $\mathbf{L}_2$-errors for the 16 test functions

and the four s2n ratios of our method (when using only standard polynomials) against the four best wavelet methods, namely W2, W3, W4, W6.

Since we found the method of Coifman and Donoho (1995) with hard thresholding (TI-hard) to be the better one, we present a more precise comparison of our results with theirs in Figure 6, in order to illustrate the influence of the choice of the filter (either the Symmlet8 filter or the Haar filter) in the wavelet methods. Our method does not require such a choice, which seems to be sometimes decisive (Signals 7 and 8).

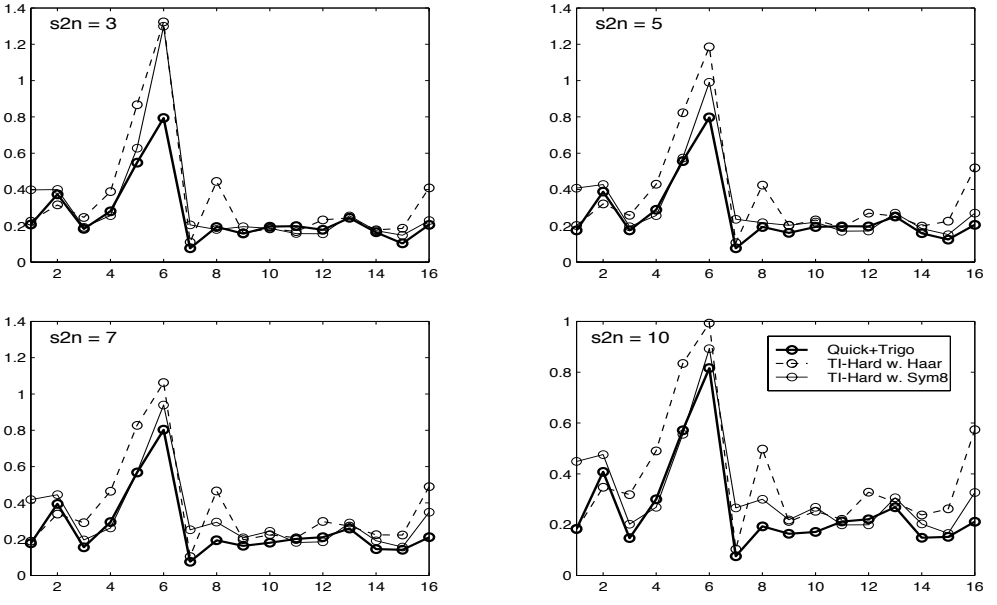

Fig. 6. Comparison of the $\mathbf{L}_2$-errors for the 16 test functions and the four s2n, of our piecewise polynomial method, using both standard and trigonometric polynomials (thick curve) and Coifman and Donoho (1995)'s method for the Haar (dashed curve) and the Symmlet8 filters (thin curve). $K = 100$, $n = 512$, $\sigma = 1$ known.

Lastly, we lead a comparison with unknown σ. We implemented our method using a preliminary estimator of σ^2 based on the mean square residuals obtained with an estimator of the regression function computed on a regular model with $D = [n/\ln(n)]$ intervals in the subdivision and degree $r = 3$. This estimator is used for the penalization procedure. The estimate of f is then used to re-evaluate σ and to initialize a second penalization. For the test functions 5 and 6, it appears clearly that almost no method makes the job in this case, neither wavelets, nor ours. The only good wavelet method is Huang and Lu (2000)'s method W5 which is never better than the other wavelet methods for the other signals. Note that the test function 5 and 6 are not used by Antoniadis et al. (2002) in their experiments. In the other cases again and as shown by the results given in Table 4, one of the better wavelet methods is still Coifman and Donoho (1995)'s method, , contrary to Antoniadis

et al. (2002)'s conclusion that the best method highly depends on the type of the signal function. Note that we gave for this method the results using both the Symmlet8 and the Haar filter.

Signal	PP/PT	W1	W2	W2H	W3	W4	W5	W6	W7	W8
1	**0.092***	2.08	0.494	*0.111*$_\bullet$	0.914	0.934	0.470	1.070	0.976	0.871
2	**0.367***	2.990	0.461	*0.385*$_\bullet$	0.946	0.960	0.510	1.030	1.200	0.871
3	**0.063***	0.209	0.111$_\bullet$	*0.120*	0.201	0.230	0.136	0.217	0.149	0.856
4	**0.206**	0.873	0.187$^*_\bullet$	*0.461*	0.274	0.305	0.240	0.215	0.365	0.857
5	**4.690**	11.4	2.57	*16.1*	2.38	2.38	0.694$^*_\bullet$	2.4	5.98	0.884
6	**23.1**	23.7	23.5	*23.6*	23.6	22.8	0.986$^*_\bullet$	23.6	23.1	4.05
7	**0.013***	0.571	0.140	*0.026*$_\bullet$	0.404	0.420	0.192	0.437	0.295	0.86
8	**0.060***	0.610	0.073$_\bullet$	*0.290*	0.243	0.260	0.275	0.259	0.402	0.864
9	**0.052***	0.390	0.081	*0.064*$_\bullet$	0.167	0.184	0.144	0.208	0.164	0.859
10	**0.090**	0.319	0.083$^*_\bullet$	*0.104*	0.128	0.154	0.120	0.136	0.182	0.859
11	**0.088**	0.204	0.053$^*_\bullet$	*0.066*	0.110	0.142	0.089	0.083	0.092	0.856
12	**0.088**	0.240	0.050$^*_\bullet$	*0.123*	0.085	0.117	0.094	0.080	0.107	0.858
13	**0.150**	0.913	0.136$^*_\bullet$	*0.181*	0.265	0.274	0.233	0.250	0.373	0.867
14	**0.053***	0.179	0.063$_\bullet$	*0.072*	0.098	0.097	0.074	0.096	0.084	0.857
15	**0.028***	0.206	0.073$_\bullet$	*0.084*	0.158	0.177	0.138	0.169	0.124	0.856
16	**0.080***	0.751	0.118$_\bullet$	*0.427*	0.239	0.258	0.256	0.234	0.362	0.868

Table 4. $\mathbf{L}_2$-errors for s2n= 5, $n = 512$, and $\sigma = 1$ is unknow. PP/PT is our method when considering both standard and trigonometric piecewise polynomials, W1 to W8 are the wavelet methods described above with Symmlet8 filter, W2H is the method W2 when using the Haar filter. $_\bullet$: best wavelet method, *: best method.

For sake of completeness, let us give a characteristic example in which our algorithm behave in a very satisfactory way, as compared with wavelets. We simulated data using the test function 3, as plotted in Figure 7.

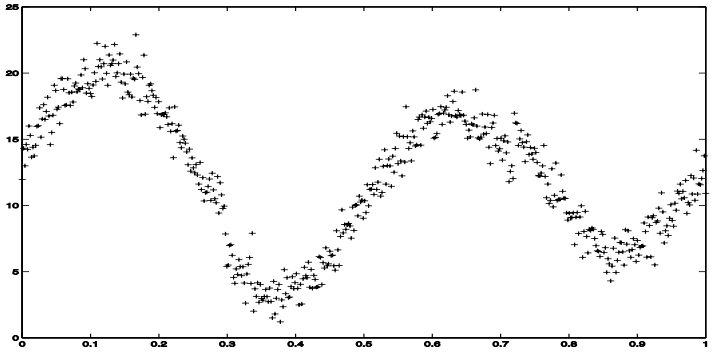

Fig. 7. Plot of $n = 512$ data simulated with function HeaviSine (test function 3), Gaussian errors, $\sigma = 1$ and s2n=5.

Then we plot in Figure 8 the true signal and the estimated function. This function presents two difficulties, a rupture and a peak. Our method detects both perfectly, some wavelet methods detect the rupture but most of them miss the peak, except W5 and W7 that give a very smooth version of it.

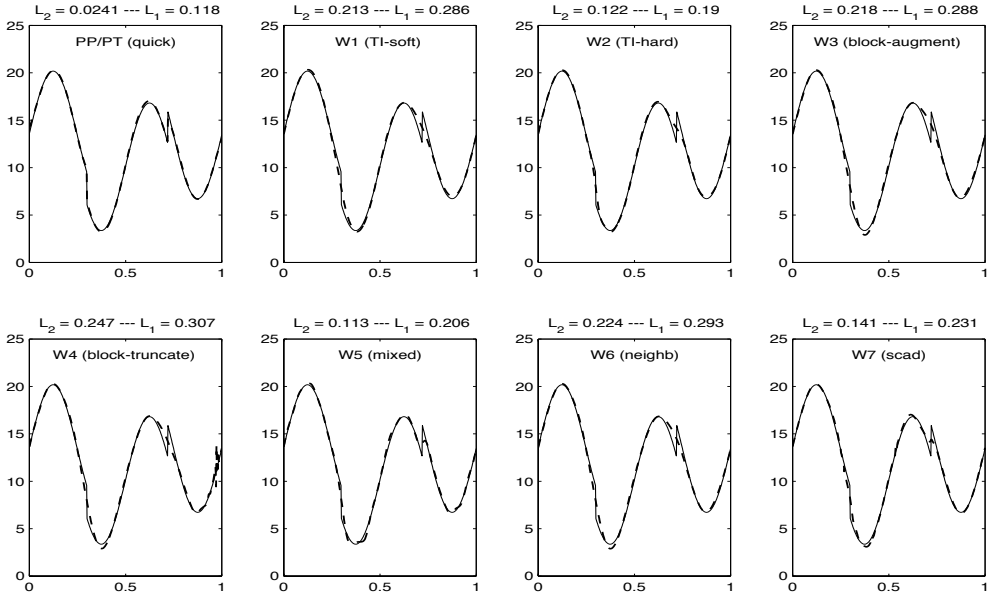

Fig. 8. True and estimated functions with methods PP/PT and W1-W7. $\mathbf{L}_2$ and $\mathbf{L}_1$ errors above the pictures. $\sigma = 1$ unknown. $n = 512$, s2n$= 5$.

6. Concluding remarks

As a conclusion, let us emphasize that we provide an algorithm that has several advantages:

(a) It is completely automatic, with no arbitrary stopping criterium as for Denison et al. (1998) method (or any MCMC method in fact), and the algorithm makes the main choices by itself. The only inputs are D_{max} and r_{max}.
(b) In (nearly) all cases, we can do as well as or better than wavelets methods; those methods remain faster, but of course, piecewise polynomials (standard or trigonometric) are much more flexible than wavelets independently of the type or signal that must be de-noised (whether square or sinusoidal or even a mix of both).
(c) As we mentioned for the complete algorithm, we have a very simple method for the actualization when the sample size increases.

(d) Lastly, our performances in term of signal compression are really excellent and should be more theoretically quantified.

Let us mention also that even if the algorithm is slower than wavelet methods, it is nevertheless quite fast. For example, compared to the one provided by Denison et al. (1998), it is ten times faster (we need 1 or 2 minutes when they require between 10 and 30).

Of course, we studied here only the simplest regression model in order to find the better algorithm to solve our problem and we still need to find some ideas to computationally deal with the general fixed design case. But several new frameworks arise immediately, which would also require a precise study.

First, we can think of working with a random design, which means replacing the deterministic x_i's by random variables X_i's

$$Y_i = f(X_i) + \sigma\varepsilon_i.$$

If the X_i's are i.i.d. and independent of the noise, then things will be right identical to the general fixed design case. Empirical experiments already prove that if the X_i's are random variables identically distributed with uniform law on $[0, 1]$ (which is the most reasonable way of generating a reasonable non equispaced design), then the estimation procedure works as well as in the fixed design case; this seems in accordance with the intuition. The only loss is in term of time. But many theoretical results are available even if they are not independent from the noise so that this assumption may also be empirically relaxed.

A simple model involving some dependency between the variables is given by the autoregressive model

$$X_i = f(X_{i-1}) + \sigma\varepsilon_i, \quad i = 1, \ldots, n.$$

Higher orders p (i.e. $f(X_{i-1})$ replaced by $f(X_{i-1}, \ldots, X_{i-p})$ with $p \geq 1$) can also be studied. Again, theoretical results (see Baraud et al. (2001b)) have been proved on that subject.

Then the ε_i's themselves may also be dependent: from the theoretical point of view, this implies some more coefficient (measuring the mixing rate of the ε_i's) in the penalty term instead of the usual σ^2. This may also be experimented, through generating dependent ε_i's, for instance built from a linear autoregression ($\varepsilon_i = \rho\varepsilon_{i-1} + u_i$ with i.i.d. u_i's).

Lastly, many experiments may be done on the variance σ^2 of the errors: it may be non constant and piecewise estimated following a specific subdivision (i.e. not the same as the one chosen to estimate the function f). More generally, heteroskedastic models are theoretically and empirically studied in the regular case in Comte and Rozenholc (2002).

References

Abramowitz, A. and Stegun, I.A.(1972) *Handbook of mathematical functions.* Dover Publications, New-York.

Antoniadis, A., Bigot, J. and Sapatinas, T. (2002) Wavelet Estimators in nonparametric regression: a comparative simulation study. *Journal of Statistical Software*, 6, i06, 2001. (see http://www.jstatsoft.org/v06/i06).

Antoniadis, A. and Fan, J. (2001) Regularization of wavelets approximations (with discussion). *J. Am. Statist. Ass.*, **96**, 939-967.

Antoniadis, A. and Pham, D.T. (1998) Wavelet regression for random or irregular design. *Comp. Stat. and Data Analysis*, **28**, 353-369.

Baraud, Y. (2000) Model selection for regression on a fixed design. *Probab. Th. Relat. Fields*, **117**, 467-493.

Baraud, Y. (1998) Model selection for regression on a random design. *Preprint of the University Paris-Sud* 97-74.

Baraud, Y., Comte, F. and Viennet, G.(2001a) Adaptive estimation in an autoregressive and a geometrical β-mixing regression framework. *Annals of Statistics*, **39**, 839-875.

Baraud, Y., Comte, F. and Viennet, G. (2001b) Model Selection for (auto-)regression with dependent data. *ESAIM*, **5**, 33-49.

Barron, A., Birgé, L. and Massart, P. (1999) Risks bounds for model selection via penalization. *Probab. Theory Relat. Fields*, **113**, 301-413.

Barron, A. and Cover, T.M. (1991) Minimum complexity density estimation. *IEEE Trans. Inform. Theory*, **37**, 1037-1054.

Birgé, L. and Massart, P. (1998) Minimum contrast estimators on sieves: exponential bounds and rates of convergence. *Bernoulli*, **4**, 329-375.

Birgé, L. and Massart, P. (2001) Gaussian model selection. *J. Eur. Math. Soc.*, **3**, 203-268.

Birgé, L. and Rozenholc Y. (2002) How many bins should be put in a regular histogram. Preprint du LPMA 721, http://www.proba.jussieu.fr/mathdoc/preprints/index.html.

Buckheit, J.B., Chen, S., Donoho, D.L., Johnstone, I.M. and Scargle, J. (1995) About WaveLab. *Technical Report*, Department of Statistics, Stanford University, USA. Available http://www-stat.stanford.edu/ wavelab.

Cai, T.T. (1999) Adaptive wavelet estimation: a block thresholding and oracle inequality approach. *Ann. Statist.* **27**, 898-924.

Cai, T.T. and Silverman, B.W. (2001) Incorporating information on neighboring coefficients into wavelet estimation. *Sankhya, Series B* **63**, 127-148.

Castellan, G. (2000) Sélection d'histogrammes à l'aide d'un critère de type Akaike. (Histograms selection with an Akaike type criterion). C. R. Acad. Sci., Paris, Sér. I, Math. **330**, 729-732.

Coifman, R.R. and Donoho, D.L. (1995) Translation-invariant de-noising. Antoniadis, Anestis (ed.) et al., Wavelets and statistics. *Proceedings of the 15th French-Belgian meeting of statisticians*, held at Villard de Lans, France, November 16-18, 1994. Lect. Notes Stat., Springer-Verlag, New York. 103, 125-150.

Coifman, R.R. and Wirkhauser, M.V. (1992) Entropy-based algorithms for best basis selection. *IEEE Trans. Inform. Theory*, **38**, 713-719.

Comte, F. and Rozenholc Y. (2002) Adaptive estimation of mean and volatility functions in (auto)-regressive models. *Stoch. Proc. Appl.*, **97**, 111-145.

Denison, D.G.T., Mallick, B. K. and Smith, A.F.M. (1998) Automatic Bayesian curve. *J. R. Statist. Soc.* **B**, **60**, 333-350.

DeVore, R.A. and Lorentz, G.G. (1993) *Constructive Approximation*, Sringer-Verlag.

Donoho, D.L. (1995) Denoising by soft-thresholding. *IEEE Trans. on Inf. Theory*, **41**, 613-627.

Donoho, D.L. and Johnstone, I.M.(1994) Ideal space adaptation by wavelet shrinkage. *Biometrika* **81**, 425-455.

Donoho, D.L., Johnstone, I.M. , Kerkyacharian, G. and Picard, D. (1995) Wavelet shrinkage: asymptopia (with discussion). *J. R. Statist. Soc.* **B**, **57**, 371-394.

Efromovich, S. and Pinsker, M. (1984) Learning algorithm for nonparametric filtering. *Auto. Remote Control*, **11**, 1434-1440.

Friedman, J.H. and Silverman, B.W. (1989) Flexible parsimonious smoothing and additive modeling. *Technometrics*, **31**, 3-39.

Hastie, T.J. and R.J. Tibshirani (1990) *Generalized additive models*. London, Chapman-Hall.

Huang, S.Y. and Lu, H.H.-S. (2000) Bayesian wavelet shrinkage for nonparametric mixed-effects models. *Statist. Sinica* **10**, 1021-1040.

Kanazawa, Y. (1992) An optimal variable cell histogram based on the sample spacings. *Ann. Statist.*, **20**, 291-304.

Krim, H., Tucker, D., Mallat, S. and Donoho, D. (1999) On denoising and best signal representation. *IEEE Trans. Inf. Theory*, **45**, 2225-2238.

Li, K.C. (1987) Asymptotic optimality for c_p, c_l cross-validation and generalized cross-validation: discrete index set. *Ann. Stat.*, **15**, 958-975.

Mallows, C.L. (1973) Some comments on C_p. *Technometrics*, 15, 661-675.

Misiti, M., Oppenheim, G. and Poggi, J.-M. (1995) The Wavelet Toolbox. The Mathworks eds.

Polyak, B.T. and Tsybakov, A.B. (1990) Asymptotic normality of the c_p test for the orthogonal series estimation of regression. *Theory Probab. Appl.*, **35**, 293-306.

Shibata, R. (1981) An optimal selection of regression variables. *Biometrika*, **68**, 45-54.

Vidakovic, B. and Ruggeri, F. (2000) BAMS method: theory and simulations. *Discussion paper*, Institute of Statistics and Decision Sciences, Duke University, USA.